

1 チップマイクロコンピュータM16の開発(1) ハードウェア記述言語を駆使した機能設計

2G-1

岩田俊一 土居俊雄 西川浩司 水垣重生 清水徹
三菱電機(株)

1. はじめに

16ビットのワンチップマイコン「M16」を開発した。M16は、32ビットのデータバスを備え、4段のパイプライン処理によりレジスタレジスタ間演算が1命令/クロックで実行可能なCPU、RAM、タイマなどの周辺機能を備えたマイクロコンピュータである。CPUのトランジスタ数は13.8万で、これを0.8 μ m CMOSプロセスにより3.7 \times 4.2mm²に集積している。

最近のプロセッサ開発においては、論理合成や自動レイアウトといった設計ツールの進歩とともに、設計効率向上を目的としてハードウェア記述言語を使用したトップダウン設計手法が導入されている。今回、M16の開発においてもCPU部の機能設計にハードウェア記述言語であるVerilog-HDLを使用することで、5名 \times 1年という小人数による短期間の開発を実現することができた。

本稿では、M16が採用したハードウェア記述言語による機能設計手法とその効果について報告する。

2. 機能設計の方針

ハードウェア記述言語は、動作モデルの作成が容易かつシミュレーションが高速であることから、機能設計や機能検証の効率化に有効である。また、ハードウェア記述言語から論理回路を自動生成する論理合成ツールを使えば、論理設計期間の大幅な短縮が可能である。しかし、実際に論理合成ツールを使う場合には、機能だけでなく回路の動作速度やレイアウトを考慮して記述しなければならない。つまり、論理合成のための機能記述は、機能設計や機能検証で要求されるよりもさらに構造的、回路的に詳細な記述が必要である。

そこで、M16 CPUの開発においては、図1に示すように、まず、機能設計と機能検証を目的とする機能モデル(=機能設計用モデル)を作成し、これを論理合成可能なモデル(=インプリメンテーション用モデル)に書き換えるという2段階に分けた機能モデルの開発を行なった。

A. 機能設計用モデル

ハードウェア記述言語であるVerilog-HDLを使用

したレジスタトランスファレベルの記述である。このモデルそのものを機能仕様書として使用する。また、このモデルを使ってテストプログラムの開発および機能検証を行う。

B. インプリメンテーション用モデル

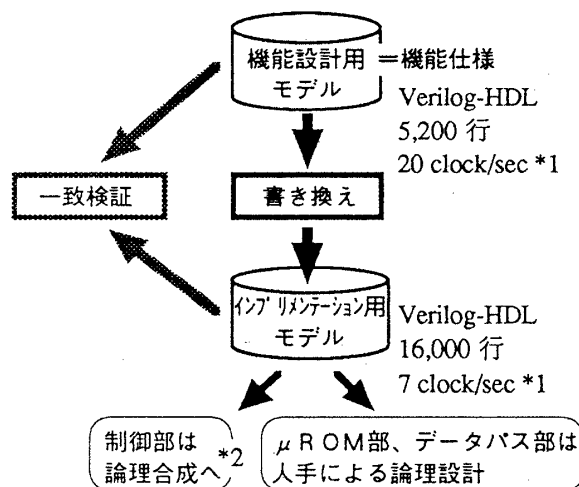
機能設計用モデルを論理合成可能な記述に書き換えたものである。ただし、論理合成を行うのは制御系のみで、データバス部や μ ROMなどは人手による論理設計を行った。機能設計用モデルとインプリメンテーション用モデルとの一致検証は、同一テストプログラムによるシミュレーション結果をクロック毎に比較することにより行った。

3. 機能モデルの記述の違い

上記2つの機能モデルはそれぞれ目的が異なっているため、記述方針も異なっている。それぞれのモデルで使用したクロックと記述フォーマットを図2に示す。

機能設計用モデルの場合は、機能仕様書としての役割を果たすため、機能的に必要なハードウェアリソースをすべて記述し、クロック単位で動作を定義した。それ以外では、早期に記述を完了して仕様書として見やすく高速なシミュレーションを達成するために、回路やレイアウトの実現方法の詳細は考慮せず、機能的な動作を明確に記述することに重点をおいた。記述上の主な特徴を以下に挙げる。

・パイプラインステージを基本にモジュール分割



*1 SPARC 370での測定値

*2 Synopsys社のDesign compilerを使用

図1 M16 CPUの機能設計フロー

M16 Single Chip Microcomputer Design (1)
Function Design Using Hardware Description Language
Shunichi IWATA, Toshio DOI, Kouji NISHIKAWA,
Shigeo MIZUGAKI, Toru SHIMIZU
Mitsubishi Electric Corporation

- ・ 1マシンサイクルを4相に分割して記述
 - ・ 各クロック毎に動作をまとめて記述
- 一方、インプリメンテーション用モデルの場合には、機能設計用モデルに対して、主に次のような箇所の書き換えを行った。
- ・ レイアウト手法を考慮してモジュールを再分割
 - ・ 4相クロックを2相クロックに整理
 - ・ 記憶素子をDラッチで明確にモデリング
 - ・ 論理合成ツール制御コマンドを付加
- このような記述方針の相違によって、インプリメンテーション用モデルの方が、記述量が3倍多く、シミュレーション速度が3倍遅くなるという結果となった。

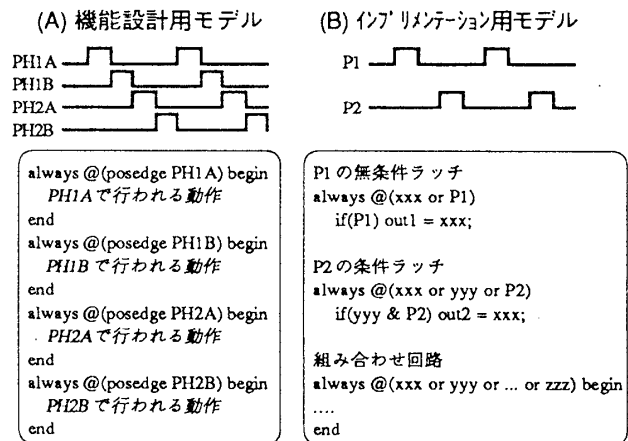


図2 M16CPUの機能モデルで使用されるクロックと記述フォーマット

4. ハードウェア記述言語を使用した機能設計の実際とその評価

機能設計開始から設計作業終了までの工程を図3に示す。

- (1) 機能設計開始

内部アーキテクチャ検討の段階で作成したプロトタイプを機能設計の出発点とした。これをもとに内部アーキテクチャ検討および機能設計が行われた。
- (2) 機能設計用モデル第1版リリース

機能設計開始から3カ月後に機能設計用モデルの第1版がリリースされた。これを用いて機能検証およびインプリメンテーション用モデルへの書き換えのトライアルと手設計部分の論理設計を開始した。
- (3) 機能設計用モデル第2版リリース

第1版に対して機能検証が80%ほど終了した時点で第2版をリリースした。第1版から3カ月後である。これをもとにインプリメンテーション用モデルへの書き換えがスタートした。機能検証は続けられた。
- (4) インプリメンテーション用モデルへの移行

インプリメンテーション用モデルへの書き換えと一致検証には2カ月を要した。一致検証が終了した時点で、機能検証対象をインプリメンテーション用モデルに移した。
- (5) 設計作業終了

機能設計スタートから約1年で機能、論理、レイアウトのすべての設計作業を終了することができ

た。この間に機能設計に携わった人数は約5名である。このような小人数、短期間の開発ができた要因をまとめる。

- ・ 機能モデルを機能設計・検証を目的とするものと論理・レイアウト設計を目的とするものに分けて開発することによって、機能仕様が早期に決定されたため、その分機能検証や論理設計を早く立ち上げることができた。
- ・ 機能仕様書にハードウェア記述言語を使用することで、事前のシミュレーションによってかなりの仕様バグを取り除くことができた。

5. 結論

M16CPU部の機能設計において、ハードウェア記述言語を用いた機能設計を行った。機能設計と機能検証を目的とするモデルを作成した後で、このモデルを書き換えて論理合成を実行できるようなモデルを作成する、という2段階に分けた機能モデル開発によって、チップの設計期間を短縮することができた。

今後はさらに自動設計の割合を増やしていくことと、論理合成用の記述への書き換えのオーバーヘッドをできるだけ少なくしていくことが課題である。

参考文献

[1] 服部ほか、「1チップCPUプロセッサの設計検証(1)(2)(3)」、第46回情処全大1993

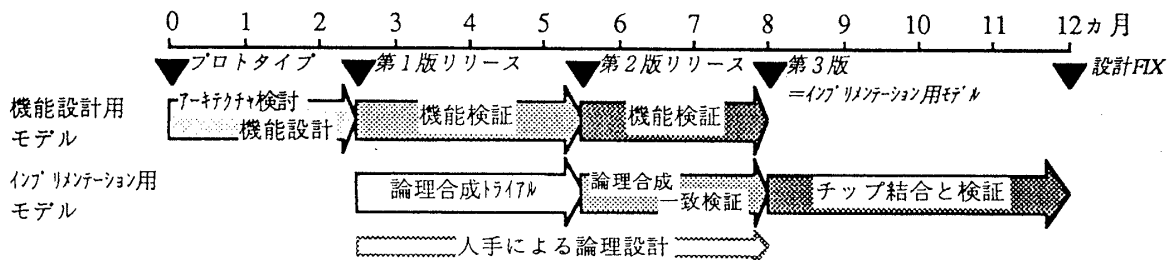


図3 M16CPUの機能モデルの開発